



# Flight Software

**Matthew Hayden**  
**Software Engineer**  
**NRL**  
**202-404-4096**  
**hayden@nrl.navy.mil**



# Flight Software Topics



**Flight Software Functional Requirements**

**Flight Software Interfaces**

**Flight Software Architecture**

**Software Development**

**Software Testing**

**Software Configuration Management**

**Flight Software Builds and Schedule**



# Bus Flight Software Re-scope Impacts



- **No Functional Changes Identified**
- **No Requirements Modified that Significantly Impact FSW Performance**



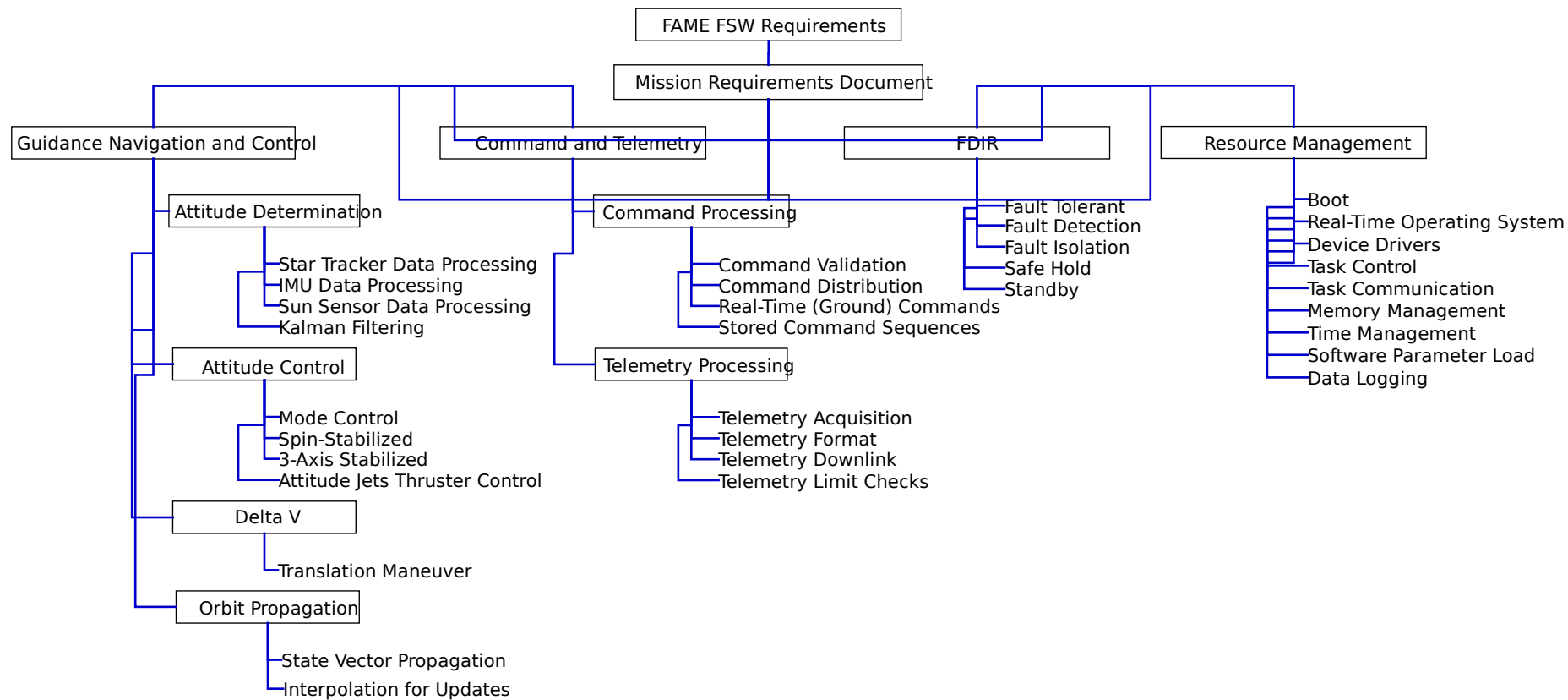
# Bus Flight Software Approach Re-Scope Impacts



- **Minimize Cost**
  - **Maximize Flight Software Code Reuse (Primarily ICM)**
  - **Eliminate non-flight (one-off) code whenever possible**
  - **Reduce documentation effort as much as possible, especially during design fluctuations**
- **Maintain Schedule**
  - **Prioritize Design and Development of Flight Software based on least likely impact from other ongoing design trades**



# Top Level Requirements





# Attitude Determination Requirements



- **Acquire initial attitude with Star Tracker (ST) quaternion output processing**
- **Process Inertial Measurement Unit (IMU) data to update attitude estimates between Star Tracker updates**
- **Process Spinning Sun Sensor (SSS) and IMU data during Apogee Kick Motor (AKM) Maneuver for Auto Nutation Control (ANC)**
- **Provide selectable Kalman Filtering of ST, IMU, and SSS for spacecraft bus attitude estimates**
- **Provide time tagged best estimate attitude to the instrument at 1 second intervals (including attitude quaternion, body rates, earth and sun vectors)**
- **Provide time tagged best estimate attitude to the ground, including current mode and validity at a commandable rate**
- **Include instrument attitude information for further refinement when available**
- **Provide sanity checks on all sensors and attitude estimates**
- **Allow for ground intervention in setting, resetting, and selecting attitude sensors and attitude information**



# Attitude Control Requirements



- **Perform seven attitude control modes**
  - **Standby, Inertial Pointing, Rate Control, Safe Hold, Open Loop Burn, Active Nutation Control, and Spin Axis Precession**
- **Monitor attitude and attitude rates for performance within defined error thresholds and report status**
- **Provide sanity checks on all commanded and autonomous attitude control actions**
- **Allow for ground intervention in setting, resetting, and limiting attitude control modes and attitude control information**
- **Log all commanded and autonomous attitude control mode changes**



# Maneuver Functional Requirements



- **Provide Spin Axis Precession and Auto Nutation Control during AKM spinup, firing, and spindown maneuver**
- **Monitor Attitude Errors and Abort Maneuver if attitude errors exceed defined thresholds**
- **Minimize probability of system reset during this mission critical phase**
- **Minimize system reset/restart time in the unlikely event of failure to retain or regain spacecraft stability within the time to effect**
- **Validate and Process Uplinked Maneuver Parameters**
- **React to real-time commands to cancel maneuver**
- **Log all executed maneuver commands and telemetry**





# Navigation Functional Requirements



- **Allow for active ranging with the ground station to be performed at commanded intervals (which denies the use of the transponder for FSW activities) without the loss of stored science data**
- **Process uplinked state vector tables and perform on-board orbit propagation calculation**
- **Provide updated state vector information as input to the Attitude Determination process**
- **Provide Sun and Earth vectors to the instrument as required**
- **Log orbit determination products**
- **Provide orbit determination products as telemetry to the ground**



# Command Functional Requirements



- **Support CCSDS Command Data Format**
- **Authenticate Received Commands (Time, Authentication Count)**
- **Validate Command Data Packets (Checksum)**
- **Distribute Real-Time Commands to Application via the Software Bus**
- **Store Time-Tagged (Absolute and Relative) and Event-Driven Commands**
- **Issue Time-Tagged Commands Based on System Time**
- **Issue Event-Driven Commands Upon Receipt of Significant Event**
- **Provide the Capability to Enable/Disable Command Sequences**
- **Log Real-Time, Time-Tagged, and Event-Driven Commands as Executed**
- **Provide Confirmation of Command Execution in Telemetry**



# Telemetry Functional Requirements



- **Support CCSDS Telemetry Data Format**
- **Acquire Spacecraft Bus Telemetry**
- **Store and Limit Check Selected TM Items Based on TM Database/Current Value Table (TDB/CVT)**
- **Monitor TDB/CVT and Generate an Event/Alarm for TM Items Exceeding Specified Limits and Persistence Counters**
- **Log All Events and Alarms**
- **Provide Sticky Indicators for Identification of Triggered Events and Errors**
- **Format Real-Time Telemetry for Downlink**
- **Time Tag Science Data for Downlink**
- **Rate Limit Telemetry Downlink Based on Selected Downlink Mode**
- **Log Selected CCSDS Telemetry Packets to Telemetry Log**
- **Load and Dump the TDB/CVT Parameters**
- **Dump Log Packets (Based on Configurable Parameters)**
- **Provide Extended Diagnostics Mode for Increased Rate Telemetry**



# Resource Manager Requirements



- **Provide a Real-Time Operating System and Application Program Interface**
- **Program Boot PROMs for Diagnostics, multiple EEPROM Software Load options, and a Minimal Command and Telemetry Capability**
- **Provide a Software Bus for Routing CCSDS Packets within FAME Applications**
- **Provide 1553 Bus Controller for RIUs and Instrument C&T Interfaces**
- **Provide Memory (Software/Data) and Object (Structured Table) Load and Dump Capability**
- **Time Management to within Instrument Specified Tolerances**
- **Interrupt Handling and Device Interfaces**
- **Memory Scrubbing (EDAC)**
- **Watchdog Timer Functions**

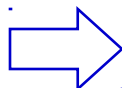


# Flight Software External Interfaces

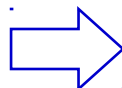


- **Primarily extra-FSC interfaces as documented in:**
  - **ICDFM004 - Spacecraft Bus to Instrument Software Interface Control Document**

Uplink Bus Commands  
Star Tracker Status/Data  
IMU Status/Data  
Sun Sensor Status/Data  
Magnetometer Status/Data  
Torque Rod Status  
Discrete Sensors Data  
Spacecraft Bus Telemetry  
Uplink Instrument Commands



**FAME  
FSW**



Downlink Bus Telemetry  
Star Tracker Control  
IMU Control  
Sun Sensor Control  
Instrument Commands  
Spacecraft Bus Commands  
Uplink Instrument Commands

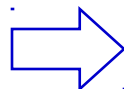


# Flight Software Internal Interfaces

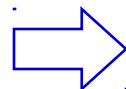


- **Primarily intra-FSC interfaces as documented in:**
  - **ICDFM005 - Spacecraft Bus to Flight Software Interface Control Document**

Timers  
Interrupts  
Events  
Ground Commands  
Stored Commands  
Object/Memory Loads  
Telemetry Collection



**FAME  
FSW**



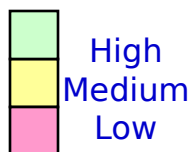
Timer Control  
Interrupt Control  
Event Management  
Task Management  
Logging  
Object/Memory Dumps  
Telemetry Reporting



# Flight Software Top-Level Architecture



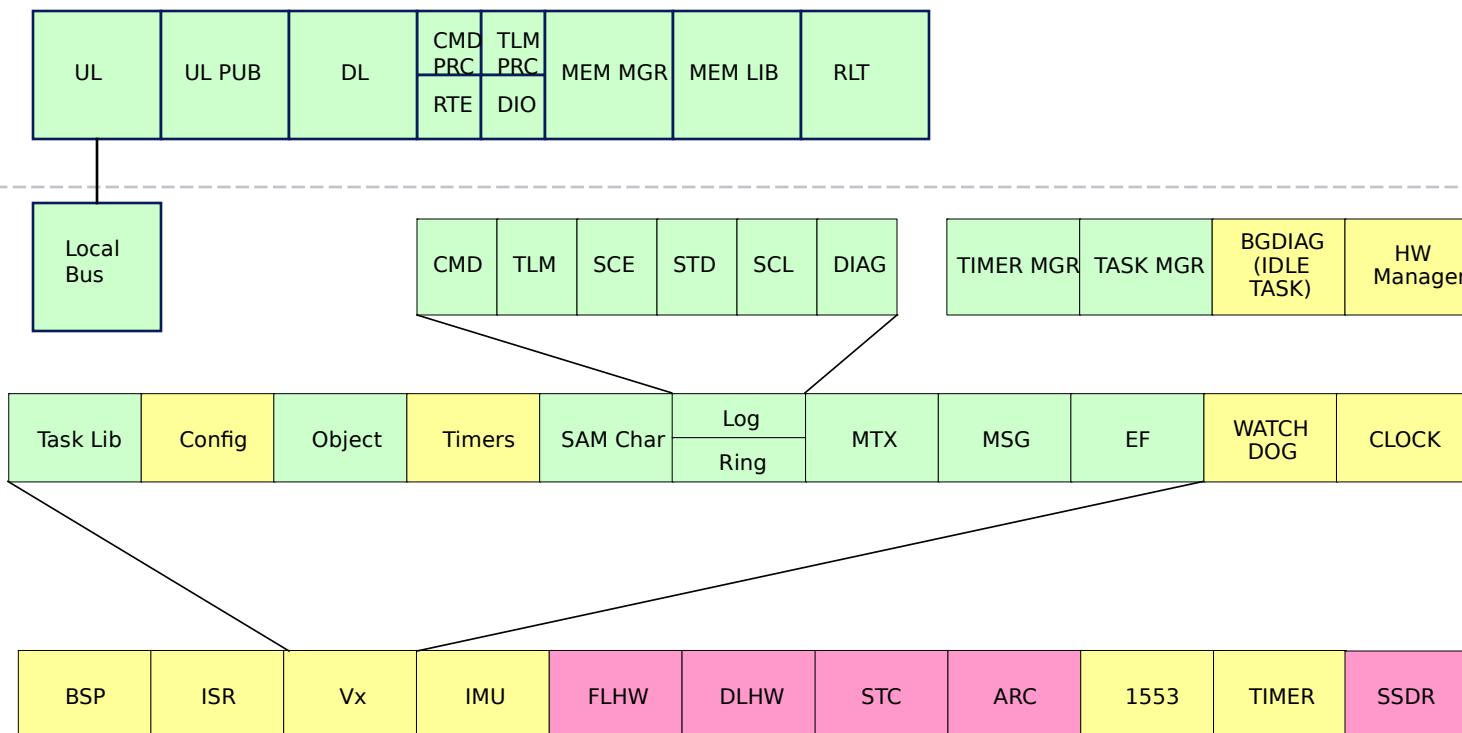
Reuse Potential



SC	NAV	ADAC	ADAC EXEC	OPP	OP
Stellar Compass					
STC ISR LIB		MATH LIB		ADAC CMD LIB	
ADAC SYNC		ADAC STS LIB		Orbit LIB	

**ADAC**

**CT**



**RM**



# Software Development Activities



- **Software Development Environment Configuration**
  - VxWorks installation
  - VxSim installation
  - ClearCase CM System
  - Restore ICM FSW test hardware where applicable
- **ICM Reuse Code Porting**
- **Uplink/Downlink design, specification, and test**
- **Device Drivers for currently designed FSC modules**
  - Uplink/Downlink Module
  - Solid State Data Recorder Module
- **Support CT&DH design**
- **TIM Participation**
- **Schedule and Budget Reviews**





# Reduced or Idled Software Development Activities

- **Bus/Instrument Communication Protocol development (reduced)**
- **Documentation Deliverables (reduced)**
- **GNC Requirement Specification (idle)**
- **GNC Design and ICM Porting (idle)**
- **Instrument Command and Telemetry Definitions (idle)**
- **Instrument Science Data Formats (idle)**





# IV&V Status



- **NASA/IV&V in Fairmont will provide IV&V program management support**
  - John Marinaro is the POC
- **SAIC/Crystal City has been identified by NASA/IV&V as the organization that will provide IV&V engineering services**
  - Paul Kirsch is the POC
- **MOA Status:**
  - Short term (until October 31st) MOA has been agreed to by NRL, GSFC & NASA/IV&V
  - NRL and GSFC signatures have been sent to NASA/IV&V, awaiting IV&V signatures and start of effort.
- **Current agreed approach**
  - Start with a short term IV&V evaluation effort (~3 months)
  - Work FAME IV&V project plan and critical function list as priorities during the initial period.
  - Preliminary review of Instrument & Bus FSW efforts will result in some initial feedback
  - Enter into long term effort that would continue until the flight software is declared ready to support the mission (revised MOA).
- **Overall Scope**
  - IV&V will only provide coverage for Instrument and Bus Flight Software Efforts



# Software Testing



- **Porting ICM Software Unit Tests as applicable**
- **Current Module, Integration, and System testing is not under active investigation or development**



# Software Documentation



## **NCST-SDP-FM001**

**Flight Software Development Plan - Currently Idle, will revise for PDR based on budget and schedule modifications**

## **NCST-SRS-FM001**

**Flight Software Requirements Specification - Currently Idle, awaiting GNC inputs**

## **NCST-ICD-FM004**

**Spacecraft Bus to Instrument ICD - Actively updated, including 1553 interface and bus schedule**

## **NCST-ICD-FM005**

**Spacecraft Controller HW to SW ICD - Actively updated, captured baseline but awaiting any CT&DH modifications**

## **NCST-ICD-FM003**

**Space to Ground ICD - Currently maintained by GSW, continuing updates of VCDU formats for supporting uplink/downlink testing**

## **FSW CM Plan**

**Updated as required, captured initial CM implementation**

## **FSW Standard Operating Procedures**

**Updated as required, captured initial FSW development procedures**

**All other documents are idle**



# Flight Software Builds



## Build 0

### System Resource Manager

- Real-OS Flash Boot
- OS APIs
- Task Manager

### Command and Telemetry

- CCSDS C&T Formatting

## Build 2

### System Resource Manager

- External Device Interfaces
- Timer Management
- Initial Boot PROMs

### Command and Telemetry

- Stored Commanding
- Limit Checking
- Instrument Data Processing

### GNC

- IMU Processing
- Attitude Determination

## Build 1

### System Resource Manager

- Breadboard H/W (FSC) Interfaces
- Object Loader
- System Logger

### Command and Telemetry

- Real-Time C&T Processing

## Build 3

### System Resource Manager

- Final Boot PROMs

### Command and Telemetry

- Event Driven Commanding
- Instrument C&T Processing

### GNC

- Star Tracker Processing
- Attitude Control



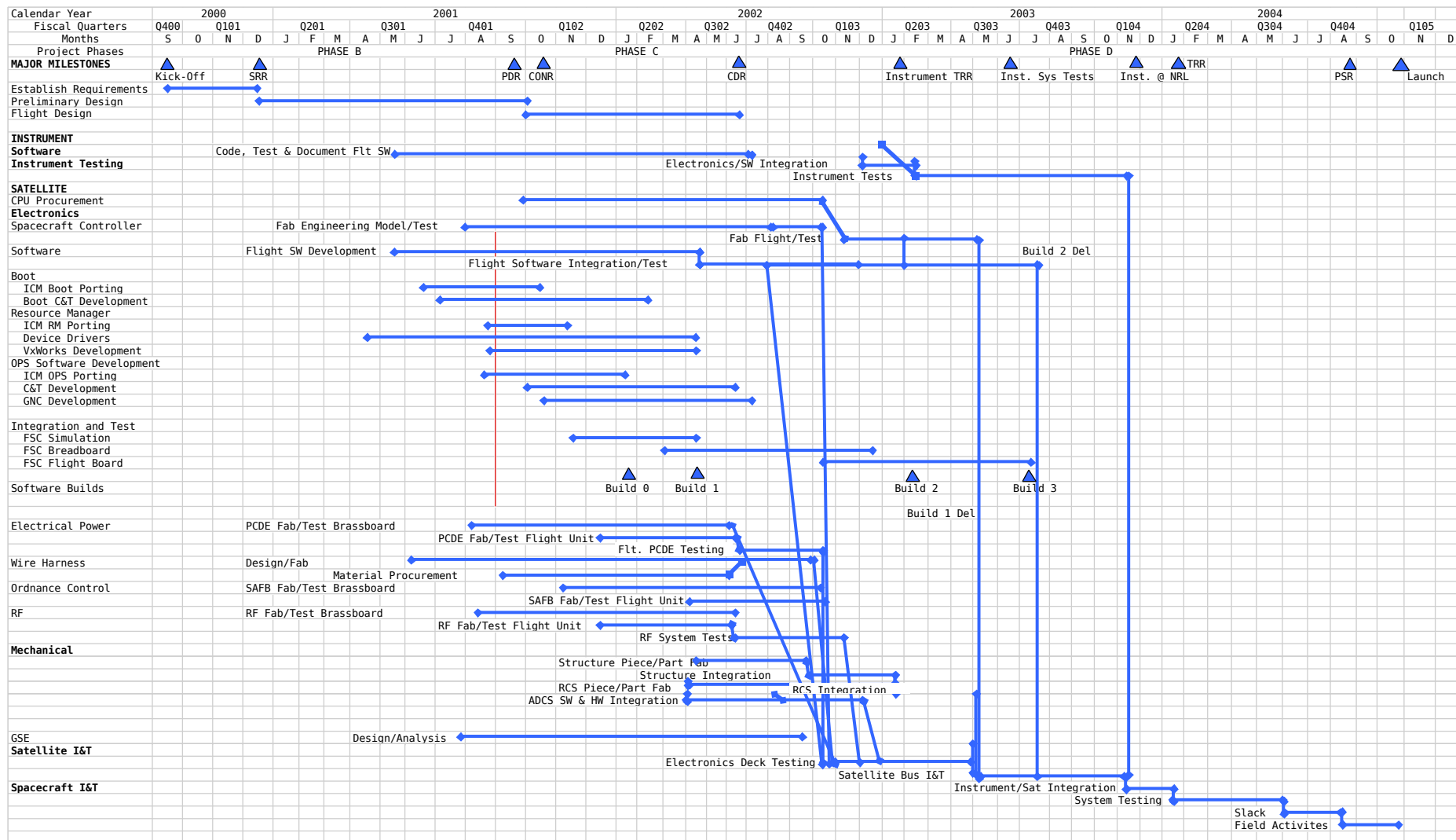
# Procurements



- **Software Development Environment Hardware and Software has already been procured**
- **Awaiting Budget Decisions regarding COMET selection**
- **Awaiting Funding regarding VxWorks Compiler Source Code Modifications**
- **Continuing Software Maintenance fees provided as needed**



# Schedule Summary





# Current Open Issues



- **Schedule was based on hardware design and delivery schedule that has been modified**
- **Testing was to be performed on hardware that may never be delivered due to budget restrictions**





# Backup

